SSARSC

Scientific Society of Advanced Research and Social Change
SSARSC International Journal of Geo Science and Geo Informatics
Volume 3 Issue 1, August 2016, ISSN 2348-6198

NoSQL Paradigm and Performance Evaluation

Pankhudi Swaroop[1], Vijit Gupta[2], Kunwar Raghvendra Singh[3], S N Rajan[4]
[1] (Corresponding Author), pankhudiswaroop59@gmail.com, Dept. of IT, IMS Engineering College, India
[2,3] Dept. of IT, IMS Engineering College, India
[4] Dean(R&D), IMS Engineering College, India

Abstract*—*Digital world is advancing by leaps and bounds thus becoming complex in terms of volume, variety and velocity. This is called the Big Data i.e. a large chunk of combined structured, semi structured & unstructured data that cannot be steered by relational databases. A new trend referred to as web2.0 is now coming into picture due to massive growth in the Internet market and surfacing of the new web technologies[1]. In order to handle these challenges the concept of NoSQL technology has been evolved as an alternative to the traditional relational database systems. This paper targets at comprehending various aspects of NoSQL, its classifications, their characteristics and evaluation of each. It aims at providing answers of the problems of handling Big Data and their analysis.

***Keywords*—NoSQL, ACID, BASE, CAP,**

II. INTRODUCTION

NoSQL that stands for "Not only SQL" was first used in 1998 by Carlo Strozzi. It is a varied yet extensively familiar set of Non Relational Database Management Systems which includes databases that are not designed chiefly on tables and generally dodge SQL for data manipulation. NoSQL database management systems are mainly used when dealing with a huge chunk of data where the data attributes can't be expressed in tables[2]. Although the SQL databases serve maximum purposes yet the only aim behind seeking a new alternative to the SQLs is that we need to excel a tool that could cover issues for which relational databases are a bad fit.

NoSQL databases flourished alongside the renowned Internet companies, such as Google, Facebook and Amazon as they too faced challenges in dealing with the large chunks of data that they were receiving every day. NoSQL systems along with providing large scale data storage and parallel processing, they also support exploratory and predictive analysis[3].

New SQL systems are relational database systems designed to provide ACID compliant, real time OLTP and conventional OLAP in big environment. To provide this New SQL uses features of NoSQL Database[4].

III. BACKGROUND

The relational database models have gained importance since 80s due to the amelioration of Oracle database, Microsoft SQL and MySQL servers. But due to the scantiness of these databases in dealing with Big Data and constraints of horizontal scalability over several servers a different approach was devised[5][6].

Trend analysis was done and out of them two of the results were considered by the International Software Community that highlighted these problems.

- The exponential growth of volume of data procreated by users, sensors and systems which was thus expanded by big distributed systems like Amazon, Google and other cloud services.
- The exceeding interdependency and intricacy of data generated by the Internet, web2.0 and social networks.

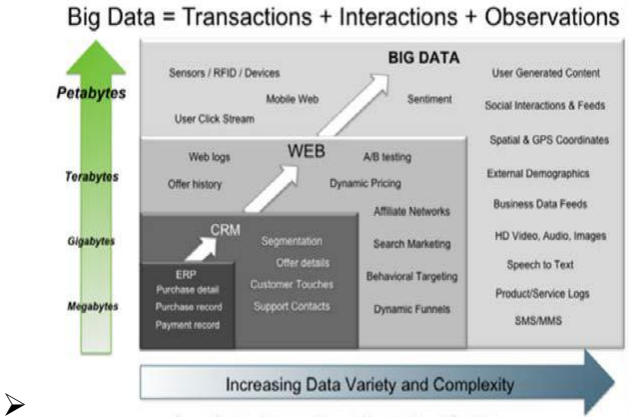


Figure 1. Big Data Transactions with Interactions

Applications such as Big Data Analytics, Business Intelligence and social networking that have computational and storage needsover petabyte have forced SQL-like centralized databases to be exploited to their limits.This led to the development of horizontally scalable, distributed non-relational data stores, called No-SQL databases, such as Google's Bigtable and its open-source implementation HBase and Facebook's Cassandra.
The featuring NoSQL databases focus on analytical processing of massive data sets, providing high scalability in chattel hardware[7][8].

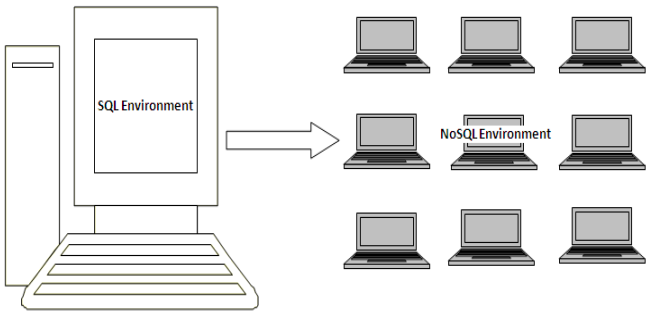


Figure 2: Decline in dominance of SQL

IV. AXIOMATICS OF NoSQL

A. ACID free

ACID is an acronym for Atomicity, Consistency, Isolation and Durability. This property is very much prominent in the SQL environment but NoSQL does not comply with the above due to the Consistency feature[9].

Atomicity: stands for 'everything or nothing'. If the transaction fails in the middle it will be completely undone.

***Consistency*:** ensures that a database before and after any transaction is stable at a valid state.

Isolation: ensures serializiabilty of concurrent transactions i.e. the processing of one transaction must not affect that of the other.

***Durability*:** ensures that once a transaction has been committed it will remain in the same state i.e. stored permanently even if there are some errors, or even if the system crash or power loss occurs.

Since in the distributed system data is spread all through the network hence any change made at a site must be visible to the users of every other sites. Each machine is responsible for the storage of its concerned data and maintenance of consistency which is a big issue with NoSQL.

B. BASE

BASE stands for Basically, Available, Soft state, and Eventual consistency.NoSQL databases travels from ACID to BASE. After a transaction consistency soft state is obtained instead of solid state. The main moto of BASE is the permanent availability factor. It provides "just in time consistency."

C. CAP

CAP stands for Consistency, Availability and Partition tolerance.

CAP works on the following three principles:

- ***Consistency:***The data in every form of its replication and updation must be available same at each site.
- ***Availability:*** Data must constantly available whenever needed.
- ***Partition Tolerance:*** Database working is not affected by device or network failure.

According to the theorem at a time only two of the properties can be satisfied. Hence selection has to be made from the pairs.

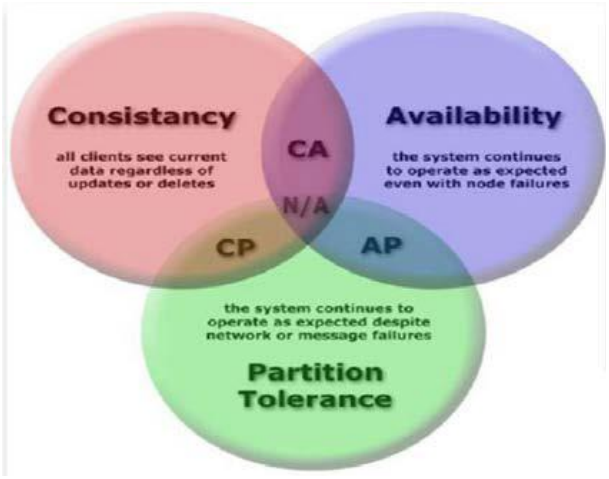


Figure 3. Characteristics of NoSQL Database (Source: nosqltips.blogspot.com)

Many NoSQL database have slacken their need for consistency and have chosen availability and partition control on their part. This gives rise to a system called BASE[9][10][11].

Han, J., Haihong, E., Le, G., and Du, J. (2011) classifies NoSQL databases according to the CAP theorem. Tudorica, B. G., &Bucur, C. (2011), compares using multiple criteria between several NoSQL databases.

Primary uses of NoSQL databases:

- Parallel processing of large scale data in a distributed environment.
- Embedded IR (basic machine-to-machine information look-up & retrieval)
- Exploratory analytics on semi-structured data (expert level)
- Volumated data storage(structured, semi structured and unstructured)

V. Classification of NoSQL Databases

NoSQL Databases are classified into four categories:

1) Key Value Stores, e.g., SimpleDB
2) Column Oriented, e.g., Cassandra, HBase, Big Table
3) Document Based, e.g., CouchDB, MongoDB
4) Graph Based

5.1 Key-Value Stores

Key-Value Stores as the name suggests is a combination of two entities: Key and Values.
It is one of the traditional databases that has given birth to all the other databases of NoSQL. It has a concrete application programming interface (API) and allows its users to store data in a schemaless manner. The stored is in two parts:

- Key is a unique identifier to a particular data entry. Key should not be repeated if one used that it is not duplicate in nature.
- Value is a kind of data that is pointed by a key.

KV Stores share similarity with the Hash Tables where the keys play the role of a unique identifier used as indexes and thus execute faster than the RDBMS. In these types of databases the only way to query is through the key, therefore they must be **unique**and arranged in an alphabetical order. To ensure greater availability the datastores are replicated.

The modern key value data stores prefer high scalability over consistency.Hence ad-hoc querying and analytics features like joins and aggregate operations have been omitted.
High concurrency, fast lookups and options for mass storage are some of the prosof key-value stores. One of the cons of key value data store is the lack of schema which makes it much more difficult to create custom views of the data.

Car	
Key	Attributes
1	Make: Nissan Model: Pathfinder Color: Green Year: 2003
2	Make: Nissan Model: Pathfinder Color: Blue Color: Green Year:2005 Transmission: Auto

Figure4. Key/Value Store NoSQL Database (Source: www.readwritewebcomimages.com)

Applications of Key Value Stores:

Key Value Databases are used to store a user's session or his shopping cart detail or details of most likely products. They are basically being used in forums, websites for online shopping. Although this kind of database stores existed long ago but they got a new evolution with the introduction of *Amazon's Dynamo.*

5.1.1 Amazon DynamoDB

Amazon DynamoDB is a newly released fully managed NoSQL database service offered by Amazon that provides a fast, highly reliable and cost-effective NoSQL database service designed for internet scale applications. It uses Amazon's Dynamo model for implementation and also offers less predictable latencies at any scale. It provides brisk access to the data since it stores data on solid state drives (SSD).The data is cloned concurrently across multiple AWS Availability Zones in an AWS Region to provide inherent soaring availability and data durability. Even under complex failure scenarios it provides high availability and durability by replicating data across at least three data centers.

5.1.2 RIAK

Riak is a distributed, fault tolerant, open source database refined by Basho technologies using C, Erlang and JavaScript. It follows the principles from Amazon's Dynamo paper and has a flexible data schema. It offers high availability, partition tolerance and persistence.

Components of Riak are Riak Clients, Webmachine, Protocol Buffers, Riak Replication, Riak SNMP/JMX, Riak KV, Riak Search, Riak Pipe and Riak Core.

Uses:

- Used in the social networking websites to provide privacy.
- Gather check out or POS data
- Regulating factory control and information system
- Building mobile applications on cloud.

Riak is used by Mozilla, AOL and Comcast.

5.2 Column Oriented Database

Column stores in NoSQL are basically hybrid row/column store unlike pure relational column databases. Although it makes use of the columnar extensions but rather storing data in the tables it stores them in extensively distributed architecture.

Columns are grouped according to the relationship of data. In column stores, each key is associated with one or more attributes (columns). A Column oriented data storestores its data in such a fashion that it can be aggregated rapidly with less I/O activity. It focuses on high scalability in data storage. The data is stored in the sorted sequence of the column family.

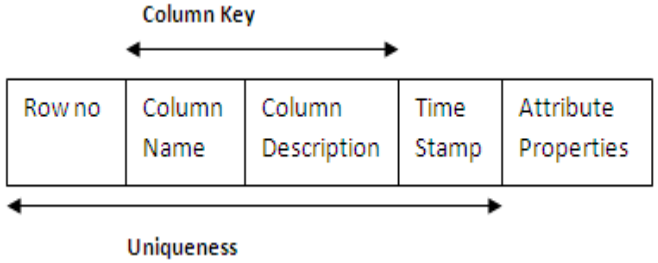


Figure5: Structure of column oriented data stores

Applications of Column Oriented Database:

(1) Widely used in distributed data storage, especially versioned data because of WC/CF time-stamping functions.

(2) Large-scale, batch-oriented data processing: sorting, parsing, conversion (*e.g.*, conversions between hexadecimal, binary and decimal code values), algorithmic crunching, *etc.*

(3) Exploratory and predictive analytics performed by expert statisticians and programmers.

(4) Suitable for data mining and analytic applications, where the storage method is ideal for the common operations performed on the data.

Figure 6. Wide-Column Store NoSQL Database

5.2.1 Big Table

Google's Big Table is acompressed high performance database which was initially released in 2005 and is built on the Google File System (GFS). It was programed using C and C++. It ensures consistency, fault tolerance and persistence. It can scale across thousands of machines and also addition of new machine to the distributed system is quite easy.

Big Table constitutes of three major pillars:

A library that is linked into every client, one master server, and many tablet servers.

Tablet servers control the working of a set of tables.

Master servers manages schema alterations, executes tasks like assigning tablets to tablet servers, balancing tablet server load, garbage collection.

Big Table is not distributed outside the Google family rather used by a number of Google applications like Gmail, YouTube and Google Earth.

5.2.2 Cassandra

Cassandra launched in 2008was developed by Apache Software Foundations. It was developed using Java. It is based on both Amazon's Dynamo model and Google's Big table. Thus it includes theories of both key-value stores and column stores. It provides variations like high availability, partition tolerance, persistence, high scalability etc. It has a dynamic schema. It can be used for a variety of applications like social networking websites, banking and finance, real time data analytics, online retail etc. Cassandra is being used by Adobe, Digg, eBay, Twitter etc. The drawback ofCassandra includes comparatively slower reads than writes.

5.3 Document Store Databases

Inspired by Lotus Notes, Document Databases were, as their name suggests, developed to manage and store documents. These databases store their data in form of documents in the databases. These documents are encoded in a standard data exchange format such as XML, JSON (JavaScript Option Notation) or BSON (Binary JSON).

Here the document are recognized by a unique set of keys and values which are almost same as there in the Key Value databases. But with a difference that the document stores become slightly more complex as they allow encasing the key-value pairs in document also known as key-document pairs.Also, unlike simple key-value stores, both keys and values are fully searchable in document databases.

Pointers are used by each filelocated in the document stores to indicate its fields.Thus, Hashing is extensively used here.Document Stores Databases are schema free and are variable in nature.

Characteristics of Document Stores Database

- Document addressing is done using unique keys in the database.
- Data can be organized as collections, tags, non-visible metadata and directory hierarchies.

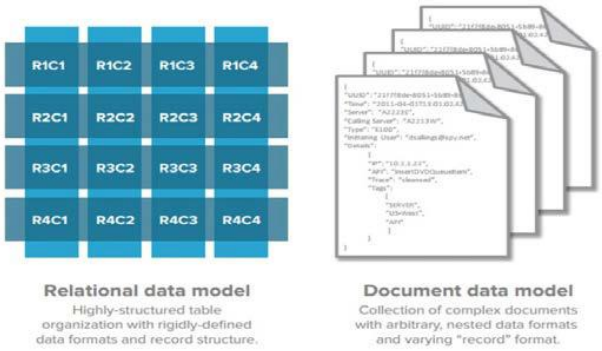


Figure 7. Document Store NoSQL Database

Applications of Document Store Database:

Document store databases are generally used in cases when the data need not be aggregated in a normalized format instead as a document with special characteristics.Document stores serve well when the domain model can be split and partitioned across some documents.They act finely in content management system, blog software etc.

They are also helpful in storing "sparse" data i.e., data that makes prominent use of nulls.

5.3.1 MongoDB

MongoDB was developed by 10gen using C++ and was initially released in 2009. It is a high performance and competent database. Features provided are consistency, fault tolerance, endurance. Some of the additional features include aggregation, ad hoc queries, indexing, auto sharding etc. In MongoDB the documents are mainly stored in BSON (Binary JSON) format. BSON documents contain an ordered list of elements consisting of field name, type and value. Also BSON is dynamic both in storage space and scan speed when compared to JSON.MongoDB uses GridFS as a specification for storing large files. It is well suited for applications like content management systems, archiving, real time analytics etc. Currently MongoDB is being used by the MTV networks, Foursquare, The Guardian etc. It is also being used in projects like CERN‟s LHC, UIDAI Aadhaar (India's unique identification project) The*drawback* is that it can be unreliable and indexing takes up lot of RAM.

5.3.2 CouchDB

CouchDB was developed by Apache software foundation and was initially released in 2005. It was developed using C++. In order to create and update database documents Couch uses JSON documents to store data and provides restful HTTPAPI. The query language incorporated is JavaScript. It provides a built in web application called FULTON which can be used for administration. It is highly available, fault tolerant and persistent. It implements Multi-Version Concurrency Control (MVCC) thus providing concurrent access to users. CouchDB supports cloning and synchronization capabilities. It can be used for applications involving occasionally changing data on which pre-defined queries have to be used. Also in cases where network connection may or may not be available, but the application must keep on working, like in the case of mobile device based applications. It can be used for CRM (Customer Relationship Management) and CMS system.

The *shortcomings* of CouchDB includesslow temporary views in CouchDB on large datasets, not good at dealing with relational data, no support for ad-hoc queries.

5.4 Graph Databases

Graph databases are databases which store data in the form of a graph. The graph consists of nodes and edges, where nodes act as the objects and edges act as the relationship between the objects. Graph databases replace relational tables with structured relational graphs of interconnected key-value pairings. The graph also consists of properties related to nodes. It uses a technique called index free adjacency i.e. every node consists of a direct pointer which points to the adjacent node. Millions of records can be traversed using this technique. In a graph database, focus is on the relation established between data using pointers. Graph databases provides schema less and efficient storage of semi structured data. The queries are expressed as traversals, thus making graph databases faster than relational databases. It is easy to scale and whiteboard friendly. Graph databases support ACID axiom and support rollback.

Applications of Graph database:

In general, graph databases are useful when relationships between data is of greater importance rather than the data itself.Graph databases find their involvement in variety of applications like social networking applications, recommendation software, bioinformatics, content management, security and access control, network and cloud management etc.

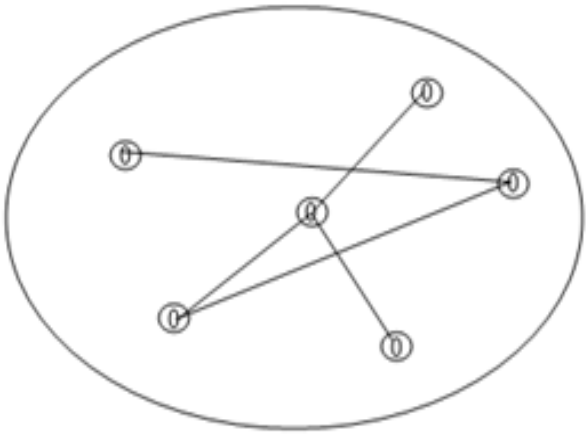


Figure8: Structure of graph database

5.4.1 Neo4j

Neo4j was developed by Neo Technology using Java and was initially released in 2007. It is a high performance graph database which ensures object oriented, flexible network structure. Neo4j supports reliability, ACID compliancy, high availability and scalability. Because of REST interface and Java API, Neo4j is quiet convenient to use. It can also be embedded into jar files. It uses CYPHER as its query language. Neo4j are generally used in software involving complex relationships like social networking, recommendation engines etc. Some of the fortune 500 companies that use Neo4j are Adobe, Accenture, Cisco, Lufthansa, Telenor and Mozilla.

VI. RELATIONAL vs. NoSQL DATABASE

A. Transaction reliability:

Relational databases follow ACID rule and hence are more reliable than *NoSQL* since they follow BASE.

B. Data Model

Relational databases store data as a set on n array relation being a subset of the Cartesian products of N domains. Data is represented as tuples and grouped into relations. The relations i.e. the table contains a set of attributes that define them as a whole.

NoSQL database follow Key-value model, Column based model, Document based model and Graph based model for representing the data. The main difference lies in the fact that NoSQL doesn't use tables for data representation also it is schema less and can handle unstructured data efficiently.

C. Scalability

Providing both vertical as well as horizontal *scalability* in the case of relational database is costly as well as practically impossible.

NoSQL database provide an effective horizontal scalability.

D. *Cloud*

Relational databases are not suitable in cloud applications the reason being inability to support full content data search hard to scale.

NoSQL databases are very much made for cloud computation itself as they work best in distributed environment.

E. *Big Data Handling*

Relational database fail to handle large chunks of data and provide no measures for their storage or analysis.

NoSQL database are devised to deal with the Big Data problem effectively.

F. *Data Warehouse*

Relational databases were designed to serve the need of data warehousing where large amount of data was being stored on the disk thus leading to Big Data problem.

NoSQL instead of focusing on data warehousing it bent on scalability and availability.

G. *Complexity*

Complexity in *Relational database* increases when data has to be stored as tables. If the structure gets disturbed at any pointprocessing could get quit complex and time taking unlike the case of *NoSQL*.

H. Crash Recovery

Relational database ensure crash recovery through its ACID properties.
NoSQL depends on replication as back up to recover from crash.

I. Security

*Relational databases*has adopted very secure mechanisms to provide the security services although they faces many security threads like SQL injection, Cross Site Scripting, Root Kits, Weak communication protocols and much more. *NoSQL*product try to solve this security issue.

We compare between relational databases security and NoSQL databases in some security services.

TABLE 1

Category	Relational databases	NoSQL databases
Authentication	All relational databases came with authentication mechanism, and can choice any of that mechanism to use.	Many NoSQL databases by default does not come with authentication or authorization mechanism, but can use some of external method to perform this operation.
Data Integrity	ACID properties that used in relational databases guarantee database transactions are processed reliably that ensure data integrate.	Eventually consistent is one of BASE properties principle therefore data integrity is not always achieved in NoSQL databases.
Confidentiality	Data confidentiality is often achieved in relational database because it was use encryption techniques, to store data encrypted.	Data confidentiality is not achieved, because usually data is store clear.
Auditing	Provide mechanisms to audit that allow writing to the database	Most of NoSQL databases don't provide auditing.

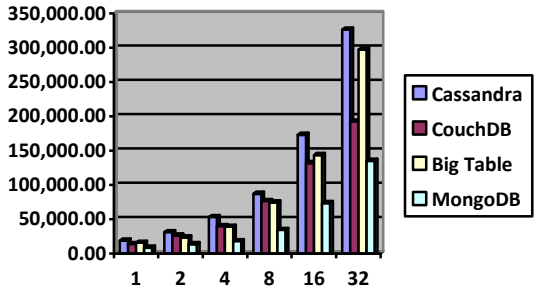
VII. PERFORMANCE EVALUATION

Talking about performance, none of the NoSQL databases can be categorized as the "winner". It is possible for one database to outperform the other when use case and deployment conditions are considered. But when other rules are applied results might completely differ.
Performance evaluation is done basically on two factors: *throughput and latency.*
NoSQL databases must provide high throughput and low latency for a wide variety of fetch-update workloads.
Scalability largely affects the throughput and latency of a system. Scalability is the ability of a system, network, or process to deal with the growing amount of work, or its potential to be enlarged in order to cover that growth. For example, it can refer to the capability of a system to increase its total output under an increased load when resources (typically hardware) are added.
There are two types of scalability:

Horizontal Scalability: This refers to adding more nodes in a system in order to deal with the increasing demands.
Vertical Scalability: This includes adding more resources to an unit in the system. These resources can be CPU or memory.

7.1 Throughput on workload:

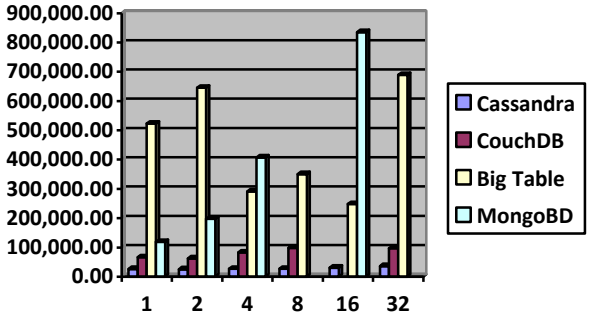
Each workload appears below with the throughput/operations-per-second (**more is better**) graphedvertically, the number of nodes used for the workload displayed horizontally, and a table with the result numbers following each graph.



From the graph we can conclude that Cassandra gives the best throughput whereas MongoDB gives the least. The trend remains same just the numerical value differs on adding new nodes (horizontal scalability).

7.2 Latency on workload:

The following latency results (**less is better**) are displayed below in microseconds.Each workload appears below with the latency graphedvertically, the number of nodes used for the workload displayed horizontally, and a table with the result numbers following each graph.



From the graph we can conclude that Cassandra gives the least throughput whereas MongoDB gives the best. The trend remains same just the numerical value differs on adding new nodes (horizontal scalability).

VII. FUTURE PROSPECTS FOR NOSQL

Although NoSQL has evolved at a higher pace still it lags behind the Relational Databases. Since the Query language being used by NoSQL is complex and users are still unaware about it. If the Query language is developed it can largely evolve the status of NoSQL in the market.

The concept of Big Data which is gaining much importance now a days can be handled using NoSQL and higher level research works can be carried out in future taking the assistance of NoSQL.

VIII. CONCLUSION

This paper covers every aspect of NoSQL database system. Starting from its classifications, we have dealt with each if its merits and demerits.

We have also dealt with comparison between the relational database and NoSQL database.

Throughput

The study depicts the pros of NoSQL which can be very well utilized if the user is skilled and also the loopholes that needs to be mended soon. If the following points are taken care of,

managing and analyzing Big Data can be made much simpler and efficient.

REFERENCES

[1] Mohamed, A, Mohamed; Altrafi, Obey; Ismail, Mohammed, "Relational vs. Non- Relational Databases- A Survey" in *International Journal of Computer and Information Technology (ISSN: 2279 – 0764) Volume 03 – Issue 03, May 2014*

[2] Robin Hecht Stefan Jablonski, University of Bayreuth " NoSQL Evaluation A Use Case Oriented Survey" 2011 International Conference on Cloud and Service Computing

[3] Dietrich Featherston "cassandra: Principles and Application" Department of Computer Science University of Illinois at Urbana-Champaign

[4] Ameya Nayak, Anil Poriya Dept. of Computer Engineering Thakur College of Engineering and Technology University of Mumbai " Type of NOSQL Databases and its Comparison with Relational Databases" International Journal of Applied Information Systems (IJAIS) – ISSN : 2249- 0868 Foundation of Computer Science FCS, New York, USA Volume 5– No.4, March 2013

[5] Eben Hewitt Apache cassandra project chair "cassandra the definitive guide" Published by O'Reilly Media November 2010

[6] Kristina Chodorow and Michael Dirolf "Mongo DB: the definitive guide" Published by O'Reilly Media September 2010

[7] http://www.mongodb.org/

[8] Fay Chang, Jeffrey Dean, Sanjay Ghemawat, Wilson C.Hsieh,Deborah A. Wallach Mike Burrows, Tushar Chandra, Andrew Fikes, Robert E. Gruber "Bigtable: A Distributed Storage System for Structured Data" Google, Inc.

[9] Lars George "Hbase the definitive guide" Published by O'Reilly Media September 2011

[10] Pokorny, J." Nosql databases: a step to database scalability in web environment" Proceedings of the 13th International Conference on Information Integration and Web-based Applications and Services. pp. 278 283. iiWAS '11, ACM, NewYork, NY, USA (2011)

[11] Elif Dede, Bedri Sendir, Pinar Kuzlu, Jessica Hartog, Madhusudhan Govindaraju Grid and Cloud Computing Research Laboratory SUNY Binghamton, New York, USA "An Evaluation of Cassandra for Hadoop" IEEE Cloud 2013